

January, 1982  
NEWSLETTER



\$1.50

# MICHIGAN ATARI COMPUTER ENTHUSIASTS

## NEW MEETING PLACE!

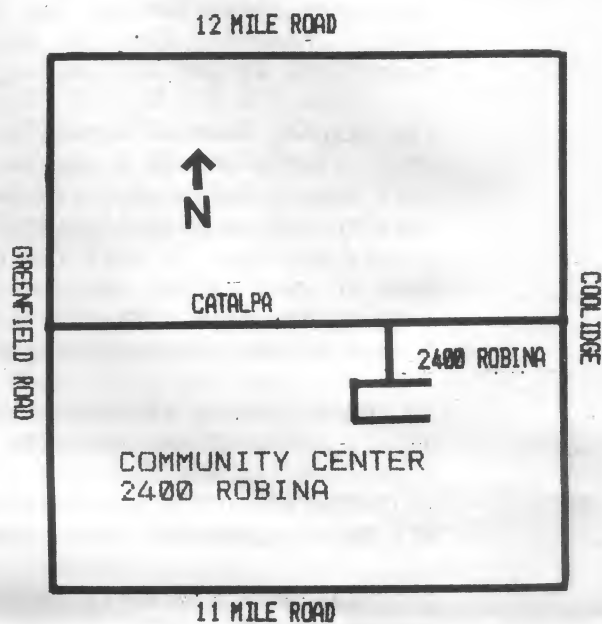
COMMUNITY CENTER  
2400 ROBINA  
BERKLEY, MICHIGAN

### MEETING AGENDA:

6:45 Open Socializing  
7:00 Libraries Open  
7:30 Call To Order  
Show and Tell  
8:00 Business Meeting  
8:30 Roundtable  
Discussions

NEXT MEETING: JAN. 21

MEETING SCHEDULE ON BACK COVER.



## IN THIS ISSUE...



RTANGLE-DEFINE A NEW DEVICE!

STRING TUTORIAL FROM  
OUR BAKER STREET DETECTIVE!

CASSETTE I/O TUTORIAL

K-RAZY SHOOT OUT REVIEW

PLUS LOTS MORE!!



LDX \$00  
LDA \$20  
STA D000,X  
STA D100,X  
STA D200,X  
STA D300,X  
STA D400,X  
STA D500,X  
STA D600,X  
STA D700,X  
DEX  
BEQ CLSE  
JMP CLA  
JSR CT  
RTS

## RTANGLE BY CRAIG CHAMBERLAIN

One of the design philosophies of the ATARI computer was that if you didn't like something about it, you could change it. In other words, it is a flexible computer. This was accomplished mainly through the use of vectors, database variables in RAM, hardware pointers and the SPECIAL command in CIO. Various interrupts jump indirectly through vectors (addresses which point to other addresses) which can be changed. Margins can be adjusted. The display memory or character set can be anywhere in memory provided you tell ANTIC where. Finally, there is room for expansion. CIO can be made to handle new operations if you are willing to create them.

The display handler normally has two SPECIAL operations. They are DRAW and FILL. Let's create a new one called RTANGLE which, as you probably guessed, will draw a rectangle. In designing this new operation we should make it as easy to use as possible, with efficiency in mind. To draw a rectangle you should not have to tell the computer all four corner points. You should only have to give it two diagonally opposite corners and the computer will take care of the rest. Before we write this new routine, however, we should take a look at how we can interface (aren't you sick of that word?) it to CIO.

In the beginning there was HATABS, a table in memory consisting only of device names (C for cassette, S for screen, etc.) and pointers to the corresponding handler vector tables. The handler tables, in turn, point to routines for CIO operations OPEN, CLOSE, and so on. Each device has its own set of routines for those operations.

Each entry in HATABS is three bytes long; one byte for the ATASCII device symbol, and two bytes to form an address which is a pointer. There are 38 bytes available to HATABS so there is a maximum of 12 different device types at any one time, with two bytes wasted. When CIO scans this table it starts looking at the end where it should find zeroes followed by entries for the various devices. It is possible to have two identical entries in HATABS but only the one later in memory will be used.

When the computer is turned on, or SYSTEM RESET is hit, HATABS is initialized to read as follows:

```
HATABS $031A 50 30 E4 P printer
              43 40 E4 C cassette
              45 00 E4 E editor
              53 10 E4 S screen
              4B 20 E4 K keyboard
```

Every entry points to a ROM address. Obviously if we want to change one of the entries in one of the handler tables we must write a similar table in RAM and then modify HATABS to point to the new table. Okay, let's take a look at a normal handler table so we can write one of our own. We are going to be writing a new operation for the display (screen) handler so that's the one we will look at.

```
DISPLA $E410 .WORD DOPEN-1  OPEN
              .WORD RETURI-1 CLOSE
              .WORD GETCH-1  GET
              .WORD OUTCH-1  PUT
              .WORD RETURI-1 STATUS
              .WORD DRAW-1   SPECIAL
              JMP PWRONA     INITIALIZATION JUMP
```

continued

## RTANGLE . continued

After you have decided where in RAM to put the new table and have written it there, write the new routine into another free place in RAM. Modify the new table to point to the new routine, but be sure to remember all vectors that you modify in case your new routine might ever want to jump to the old routine. Now modify HATABS to point to your new vector table and you are in business.

I placed both the new table and routine on page \$06. The new routine checks the command byte to see if it was a \$13 (remember that \$11 and \$12 are used for DRAW and FILL). If not, it jumps to the old routine. Otherwise it manipulates the four co-ordinates and draws the four legs of the rectangle. It can be called from ATARI BASIC by setting the COLOR, doing a PLOT to one point, doing a POSITION to the diagonally opposite corner, and then doing an XIO 19, #6, 0, 0, "S:" .

Any operation which does not classify as OPEN, CLOSE, GET, PUT, or STATUS is considered SPECIAL. Command numbers 1 through 13 are reserved for the five primary operations so a command number greater than 13 or \$0D indicates SPECIAL. The routine handling SPECIAL must then decide what to do with the command. Some devices such as the keyboard have no SPECIAL function.

Notice that all of the addresses in the handler table are actually one less than the true address. That is because these addresses were intended to be pushed onto the hardware stack. A 6502 RTS (return from subroutine) instruction executed now would cause the two address bytes to be pulled from the stack and placed into the 6502 program counter, thus fooling the computer into thinking that the last instruction it executed was at address minus one. The next instruction will come from the following location, which is address minus one plus one, which is exactly where you wanted to be. It's just a tricky way to do a JMP on the 6502. If you don't understand it don't worry about it.

Back to the original problem of drawing a rectangle. The first step is to write a handler vector table to be placed in RAM. Be careful, here, because although the ROM location of the default table at DISPLA is guaranteed not to change with future versions of the operating system, it is possible that the addresses for the individual CIO operations might change. Also it is possible that when the new table is installed another one also in RAM for the same device might already have been in use. Rather than copying the ROM table into RAM, search HATABS to find which table it is currently using for that device and copy that one. In this way you will avoid conflicts between different operating system versions and pre-modified handler tables.

Here are a couple of technical notes. You may have discovered that when using the FILL operation in the immediate mode the filled area is whatever color you had put into FILDAT, \$02FD, but the left edge is in COLOR 3. That's because when you type a line in you end it with a carriage return, which is character number 155, which is the same as COLOR 3. You will encounter this problem with RTANGLE but still it only happens in the immediate mode. The solution to that is to either do a PLOT, DRAWTO, etc. command on the same line before the RTANGLE or directly POKE ATACHR (\$02FB or 763) with the correct COLOR value. continued

## RTANGLE continued

Remember that it is possible to POSITION out of range but RTANGLE will generate an ERROR 141 if you try to draw a rectangle out of range.

A few who know assembly language may be interested in how I wrote the routine. I chose to use the stack rather than reserve locations to hold co-ordinate data. I saved memory in two ways; I used no extra locations and I saved two bytes every time I did a PHA rather than a STA. I could have used some spare bytes but you never know what else they might be used for. Using the stack did create one problem, though. If there was an error I had to pull the unused bytes off the stack.

RTANGLE is not a very practical function. The primary purpose of this article was to demonstrate how to add new operations to CIO and clear some of the mystery surrounding vectors, device tables, and all stuff. I hope I have accomplished my goal. Have fun with RTANGLE and be thinking of other operations you would like to have implemented. Happy holidays...

*Craig Chamberlain 11/27/81*

```
1000 REM RTANGLE 11/27/81 by Craig Chamberlain (313) 642-0963
1010 RDISP=1536:REM reserve area on page $06
1015 REM for new display handler vector table
1020 SSPECL=RDISP+15:REM place new display handler SPECIAL routine after
1030 REM new vector table
1040 HATABS=794:REM device handler table
1050 REM search HATABS for current display handler entry
1060 K=HATABS+36
1070 REM 83 is ATASCII for "S"
1080 IF PEEK(K)=83 THEN 1120
1090 REM each entry in HATABS is three bytes long
1100 K=K-3:GOTO 1080
1110 REM find current display handler table
1120 DISPLA=PEEK(K+1)+256*PEEK(K+2)
1130 REM read from current table into RDISP
1140 FOR J=0 TO 15:POKE RDISP+J,PEEK(DISPLA+J):NEXT J
1150 REM note that all vector table entries
1155 REM point to one less than their true address
1160 SSPECL=SSPECL-1
1170 REM change the SPECIAL entry to point to SSPECL
1180 SSPECLH=INT(SSPECL/256)
1190 POKE RDISP+10,SSPECL-256*SSPECLH:POKE RDISP+11,SSPECLH
1200 REM restore SSPECL
1210 SSPECL=SSPECL+1
1220 REM read in new SPECIAL routine
1230 FOR J=SSPECL TO SSPECL+75:READ N:POKE J,N:NEXT J
1240 REM modify HATABS display handler entry to point to RDISP
1250 RDISPH=INT(RDISP/256)
1260 POKE K+1,RDISP-256*RDISPH:POKE K+2,RDISPH
1270 END
1500 DATA 165,34,201,19,240,9,173,27,228,72,173,26,228,72,96,169,17,162,96
1510 DATA 157,66,3,165,84,168,165,90,72,133,84,165,91,72,165,92,72,152,72
1520 DATA 32,86,228,48,26,104,133,84,32,86,228,48,19,104,133,86,104,133,85
1530 DATA 32,86,228,104,133,84,32,86,228,76,52,246,104,104,104,104,76
1540 DATA 58,246
```

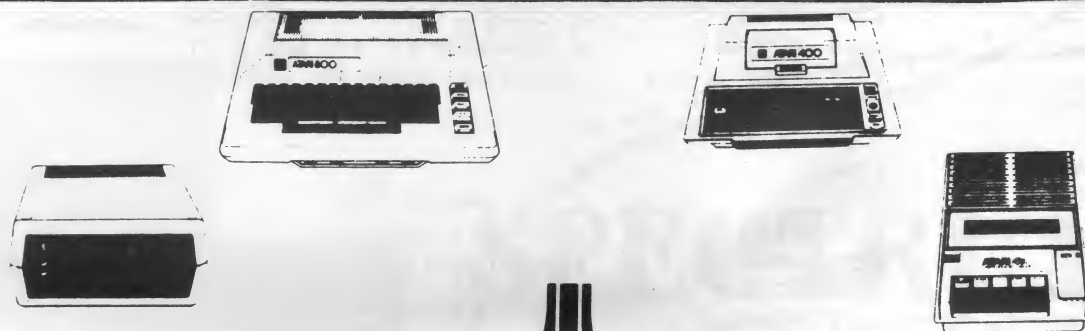
*continued*



|      |        |      |                                   |                                    |
|------|--------|------|-----------------------------------|------------------------------------|
| 0632 | 48     | 1520 | PHA                               |                                    |
| 0633 | 98     | 1530 | TYA                               | ; get ROWCRS back                  |
| 0634 | 48     | 1540 | PHA                               | ; ROWCRS pushed last, pulled first |
| 0635 | 2056E4 | 1550 | JSR CIOV                          | ; DRAW first leg                   |
| 0638 | 301A   | 1560 | BMI ERR1                          |                                    |
| 063A | 68     | 1570 | PLA                               | ; get Y2                           |
| 063B | 8554   | 1580 | STA ROWCRS                        |                                    |
| 063D | 2056E4 | 1590 | JSR CIOV                          | ; DRAW second leg                  |
| 0640 | 3013   | 1600 | BMI ERR2                          |                                    |
| 0642 | 68     | 1610 | PLA                               | ; get X1                           |
| 0643 | 8556   | 1620 | STA COLCRS+1                      |                                    |
| 0645 | 68     | 1630 | PLA                               |                                    |
| 0646 | 8555   | 1640 | STA COLCRS                        |                                    |
| 064B | 2056E4 | 1650 | JSR CIOV                          | ; DRAW third leg                   |
|      |        | 1660 | ; any range errors will have been |                                    |
|      |        | 1670 | ; detected by now so don't bother |                                    |
|      |        | 1680 | ; with further error checking     |                                    |
| 064B | 68     | 1690 | PLA                               | ; get Y1                           |
| 064C | 8554   | 1700 | STA ROWCRS                        |                                    |
| 064E | 2056E4 | 1710 | JSR CIOV                          | ; DRAW last leg                    |
| 0651 | 4C34F6 | 1720 | JMP RETUR1                        |                                    |
| 0654 | 68     | 1730 | ERR1 PLA                          | ; exit early so                    |
| 0655 | 68     | 1740 | ERR2 PLA                          | ; must pull extra stack bytes      |
| 0656 | 68     | 1750 | PLA                               |                                    |
| 0657 | 68     | 1760 | PLA                               |                                    |
| 0658 | 4C3AF6 | 1770 | JMP RETUR1+6                      | ; show error in Y                  |
| 065B |        | 1780 | *= \$0324                         |                                    |
| 0324 | 0006   | 1790 | .WORD RDISP                       | ; modify screen handler entry      |
|      |        | 1800 | ; in HATABS                       |                                    |

**NOTE:** Every time SYSTEM RESET is pressed HATABS is re-initialized.  
In this case you must modify HATABS again before running the program.

# Computers for people.™



**COMPUTER CONNECTION, INC.**

38437 GRAND RIVER • FARMINGTON HILLS, MICHIGAN 48018

Tuesday, Wednesday & Saturday 10:00 - 6:00 / Monday, Thursday & Friday 10:00 - 8:00

(313) 477-4470

RTANGLE 11/27/81 by Craig Chamberlain (313) 642-0963

```

0000      1010      .PAGE
          1020 ; new SPECIAL command for display handler
          1030 ; to draw a rectangle given two
          1040 ; diagonal points
          1050 ; to use do PLOT, POSITION, then
          1060 ; call CIO 19 to #6 ("S:")

E456      1070 CIOV   =   $E456      ; Central Input/Output vector
0342      1080 ICCOM  =   $0342      ; command byte also known as ICCMD
0022      1090 ICCOMZ =   $22        ; zero page command byte from CIO
0054      1100 ROWCRS =   $54        ; current cursor Y position
0055      1110 COLCRS =   $55        ; current cursor X position
005A      1120 OLDROW =   $5A        ; former cursor Y position used
          1130 ; used by DRAW and FILL
005B      1140 OLDCOL =   $5B        ; former cursor X position
E410      1150 DISPLA =   $E410      ; display handler vector table entries
          1160 ; note that these entries may change
F3F6      1170 DOPEN  =   $F3F6      ; OPEN
F634      1180 RETUR1 =   $F634      ; CLOSE and STATUS
F593      1190 GETCH  =   $F593      ; GET
F5B7      1200 OUTCH  =   $F5B7      ; PUT
FCFC      1210 DRAW   =   $FCFC      ; SPECIAL
F3E4      1220 PWRONA =   $F3E4      ; INIT
0000      1230      *=   $0600      ; start on page $06
          1240 ; new vector table
0600 F5F3      1250 RDISP .WORD DOPEN-1
0602 33F6      1260      .WORD RETUR1-1
0604 92F5      1270      .WORD GETCH-1
0606 B6F5      1280      .WORD OUTCH-1
0608 33F6      1290      .WORD RETUR1-1
060A 0E06      1300      .WORD SSPECL-1
060C 4CE4F3     1310      JMP PWRONA
          1320 ; new SPECIAL handler
060F A522      1330 SSPECL LDA ICCOMZ      ; get the command byte
0611 C913      1340      CMP #$13        ; check if it is RTANG
0613 F009      1350      BEQ RDRAW      ; branch if it is
0615 AD1BE4     1360      LDA DISPLA+11    ; else jump to DRAW
0618 48        1370      PHA
0619 AD1AE4     1380      LDA DISPLA+10
061C 48        1390      PHA
061D 60        1400      RTS
061E A911      1410 RDRAW LDA #$11        ; do weird JMP
0620 A260      1420      LDX #$60        ; DRAW command
0622 9D4203     1430      STA ICCOM,X      ; prepare for I/O to the screen
0625 A554      1440      LDA ROWCRS
0627 AB        1450      TAY              ; save Y2 for later use
0628 A55A      1460      LDA OLDROW
062A 48        1470      PHA              ; save Y1 on stack
062B 8554      1480      STA ROWCRS      ; set Y1 for first DRAW
062D A55B      1490      LDA OLDCOL
062F 48        1500      PHA              ; save X1
0630 A55C      1510      LDA OLDCOL+1    ; X position requires two bytes

```

*continued*

# January Clearance Sale!

All Books **20% off** Most Software **20% to 50% off**

Additional Savings on Selected Items

Sale ends January 30

# SPECTRUM

## THE COMPUTER STORE

### WE OFFER:

- \* A sales staff that gives personal service—and will answer all your questions.
- \* Up-to-date selection of business and personal software.
- \* A service department that cares for your equipment—in store or on site.



apple computer



- \* Complete line of computers, accessories, supplies featuring:
  - Altos Hard Disk/multi-user
  - Apple Computer Systems
  - Atari Computers

### HOURS:

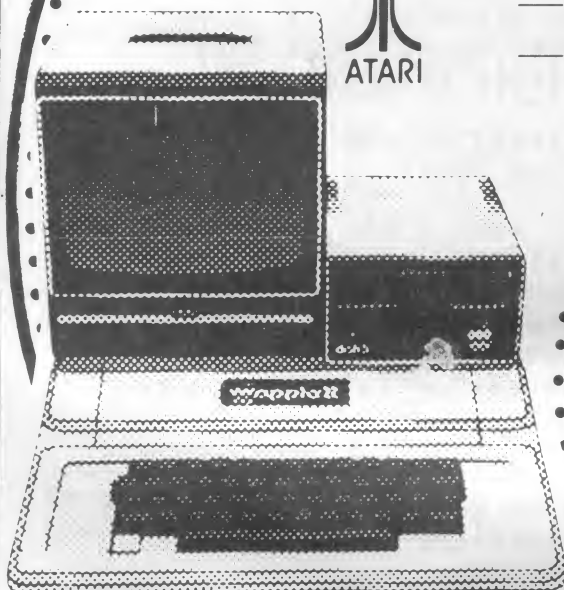
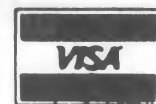
M-F 10 - 7:30 p.m.

Sat. 10 - 6:00 p.m.

## SPECTRUM COMPUTERS

26618 Southfield Rd.  
Lathrup Village, MI 48076  
(Between 10 & 11 Mile)

Phone: 552-9092



## CORRESPONDENCE

Editor  
M.A.C.E. Newsletter

October 9, 1981

Dear Sir:

Hi. I'm Bill Wilkinson, a member of the software team that produced Atari BASIC and, of course, our own BASIC At (which quite a few of your members have, apparently). I wanted to comment on Crais Chamberlain's generally very good article ("Hyperware your Atari BASIC Program", September, 1981).

I. In his paragraph entitled "Effectively Use the Computed GOTO Statement", I think even his final version code can be improved for most cases. His code:

a. IF P<>BARR AND P<>ROBOT THEN ON (P=PLAYER)+1 GOTO 280,600  
Since I don't know what values BARR, ROBOT, and PLAYER have, I will arbitrarily assign PLAYER=1, BARR=2, and ROBOT=3.

Since P can only have the values of 0, 1, 2, or 3 (because it is the result of LOCATE X,Y,P), we could simply code

b. ON P+1 GOTO 280,600,270,270 .

This is reasonably efficient, but takes a little long in the case of P=ROBOT. A quick improvement is

c. ON P+1 GOTO 280,600

because the ON will "fall thru" (just like IF) if the expression (here, P+1) is less than 1 or more than the number of line number expressions. This works because we assumed that the values we wished to branch on were 0 and 1 (if they had been 2 and 3, we could code "ON P-1 ..." -- in fact, this can work for any pair of adjacent values). But suppose PLAYER=2. Then we might code

d. ON (P=0) + 2 \* (P=PLAYER) GOTO 280,600 .

Tricky, but still more efficient than the IF combined with ON; especially because if P=BARR or P=ROBOT the expression is zero, and the ON code falls thru quicker. The best thing about this version is that it works for any pair of values selected from a list of any number of items.

Finally, if you are worried about speed in the cases of P=BARR or P=ROBOT, the most direct way is the best:

240 LOCATE X,Y,P : IF P=0 THEN 280

250 IF P=PLAYER THEN 600

Reason: fall thrus are quick, expressions are slow, GOTO can be slow. This last version evaluates only two expressions before setting to line 270. Since line numbers are expressions, the versions above generate the equivalent of the following number of expressions:

a: 5 ; b: 5 ; c: 3 ; and d: 3+ (expression is more complex)

A postscript: when code space is of prime concern, consider the terribly unstructured but highly memory-efficient GOTO <expr>. This works especially well when you have a large number of branches and you can put the destinations on lines that are evenly spaced:

240 GOTO 250+10\*P

250 rem case of P=0

260 rem case of P=1

270 rem case of P=2

...



II. I disagree about using REM lines as the object of GOTOs and GOSUBs. It is extraordinarily GOOD programming practice to use REMs as the start of each block of code; it is so much easier, then, to insert missing lines, put in tracing statements, etc. A REM executes extremely fast, so I would not take REMs out until I got desperate for space or speed. (Perhaps this is a holdover from my assembly language training. And did you know that our EASMD and, naturally, the Atari assembler cartridge allow this:

LABEL

LDA (zz),Y ; the point is that the LABEL is on the line  
; all by itself. VERY, VERY good practice! )

III. Unlike virtually any other microcomputer BASIC (perhaps you should read that as "unlike Microsoft BASIC"), Atari BASIC and BASIC At do NOT, repeat NOT, evaluate numeric literals at run-time!

POKE K,127

and

POKE K,P

execute in almost the same time (in fact, the version with the constant executes just a little FASTER...NOT slower). With Atari BASIC, use variables instead of constants to save space, NOT to gain speed.

IV. A minor point not directly related to Craig Chamberlain's article, but pertinent: the programette (a new computerese word!), titled "Inverting a String", used "...ASC(X\$(I,I))...". Since the ASC function only looks at the FIRST character of the string it is given, a little space and time can be saved by writing that as "...ASC( X\$(I) )..."

V. Finally, for something which really has nothing to do with the subject of saving time or memory, add the following lines to the "Fill In A Shape" program by Lane Winner:

1001 SOUND 0,F,10,15 : F=F+1 : IF F > 255 THEN F=0

1129 SOUND 0,F,10,15 : F=F-1 : IF F < 0 THEN F=255

Now, listen as the program runs. The pitch goes down as GOSUBs are executed and up as RETURNS are executed (it recycles if it gets too high or too low for SOUND). How many levels of GOSUB nesting can you count? (It's easy to estimate, multiply the number of recyclings by 256.) Hmmm?? Don't bother trying this with most BASICs; it won't work. We allowed GOSUB/RETURN and FOR/NEXT (and WHILE/ENDWHILE in BASIC At) to nest as deep as memory permits.

By the way, you can't even code this program in 6502 assembly language! Unless you play with the stack a whole bunch.

That's about all I have to comment on for now. I must say I was favorably impressed with your newsletter; it is definitely one of the better ones. Good luck and keep up the good work.

William F. Wilkinson

COMMENTARY: CRAIG CHAMBERLAIN TAKES A GIANT STEP BACKWARDS  
(WITHOUT EVEN ASKING "MOTHER MAY I")

In reference to Bill Wilkinson's letter I must admit that I really blew it this time. He is totally correct in that I did not use the most efficient coding possible. He got me again when he pointed out that ATARI BASIC does not evaluate numeric literals at run time. I found out about that when I read the DE RE MANUAL, after I wrote the article. I did call Bill and found him extremely nice to talk to. He seemed to agree with me that my error was a case of lack of information. Quite frankly, ATARI BASIC does some things very differently than any other BASIC I have ever dealt with. It's unfortunate that it takes a jerk like me to write something wrong in order to find out about these differences. It would have been nice if ATARI had told us users about some of these differences. Anyways, the DE RE MANUAL does contain many more speed and memory saving suggestions, so read it. As to remark statements, I still disagree. I would put remarks right before or after the addressed line. That way, if the remarks are ever removed, you will not have undefined GOTO statements. Finally, the nesting feature of subroutines and loops is something I'm sure we all appreciate. *Craig*



36575 MUDGE RANCH ROAD  
COARSEGOLD, CALIFORNIA 93614

PHONE (209) 683-6858

M.A.C.E.

Jerry Aamodt  
30500 Van Dyke Ste. 204  
Warren MI 48093

Dear Mr. Aamodt,

In answering your letter of November second, asking us to advertise in your newsletter, I can only be surprised! Our products are already advertised in your newsletter, or actually "unauthorized" copies of our product are already advertised in your newsletter. I am speaking of SPECTRUM computers ad selling JAWBREAKER on cassette. Since you no doubt know our product, you know that JAWBREAKER has not yet been released on cassette.

I judge a newsletter by its advertisers and yours are obviously not reputable.

I have read your newsletter, as well as those of other users groups, faithfully since this company entered the ATARI market months ago. I have been impressed with your letter, but company policy does not permit me to advertise with any publication that offers advertising to nibble or bit copy programs, known pirates or copyright violaters. ON-LINE SYSTEMS is one of many software publishers that adheres to this type of policy. I can only hope that you seek to rid your newsletter of this type of advertiser so that established software companies may once again look at your publication as a sales vehicle for software product.

Regards

John Williams  
Marketing Manager

## M.A.C.E. Replies to On-Line:

Michigan Atari Computer Enthusiasts  
P.O. Box 2785  
Southfield, MI 48037

December 1, 1981

Mr. John Williams  
On-Line Systems  
3675 Mudge Ranch Road  
Coarsegold, CA 93614

Dear Mr. Williams:

As secretary of MACE, I would like to respond on behalf of the group to your recent letter to Jerry Aamodt concerning the advertising policies of the MACE newsletter. From the start, our members have agreed that MACE would stay clear of the type of trading of commercial software that goes on in some users' groups. It has always been our policy to screen submissions to the club's software library and to the newsletter, so as to avoid the direct violation of copyrights by MACE as a group, or the encouragement of such violations by its individual members.

We do not believe that by running the ad referred to in your letter we have violated that policy. Spectrum Computers has in no way been engaged in the unauthorized duplication and sales of your product on cassettes. Rather, they have been selling only authorized copies of Jawbreaker on disk, which have been obtained from your company. They have then offered technical assistance to their customers, and helped them to create a single archival copy on cassette of the software which they have just purchased.

MACE recognizes that the subject of archival or "backup" copies is a sensitive one among software publishers. Nonetheless, as a group of software users, we are well aware of the legitimate need to maintain such copies of valuable software. This need is so clear that Congress made the right of a purchaser of software to maintain an archival copy of that software an explicit part of the new copyright law. What our advertiser is doing is consistent with this law, and has been increasing sales of your product. The people at Spectrum have handled their policy of technical assistance to cassette owners responsibly, and have neither distributed "copy programs" indiscriminately, nor in any way encouraged violation of the copyright laws.

In light of this, we don't feel it necessary to reject their advertising simply because their actions are contrary to the obvious preference of the software producer to keep his product in a "protected" format, regardless of how that format may lessen the software's utility to the consumer (my own copy is so well "protected" that it won't boot up half of the time). Although we at MACE like to think of ourselves as a responsible organization, we cannot claim the requisite moral superiority to tell our

members, most of whom do not own a disk drive, that if they pay thirty dollars for Jawbreaker on disk, and have a legitimate license to use that software, it is wrong for them to create an archival copy on cassette for their own use and enjoyment, because it will tempt them into making unauthorized copies for distribution to others in violation of the publisher's rights. Since we are not an association of software publishers, we cannot easily assume such a predisposition to crime on the part of our members.

We do agree that the Spectrum ad is worded in a manner which might convey the impression that Spectrum is dealing in unauthorized software with the approval of MACE, and will ask our advertisers to be more careful in the future. You must remember, however, that our newsletter is of limited circulation, and is primarily aimed at our local members, who are familiar with both MACE and its advertisers, and are thus in a position to judge both by their actions, as well as by their words.

I should add that On-Line's criticism of MACE might have more impact if its own moral and legal position were beyond reproach. You, better than most people, should realize that there are many grey areas of copyright law, and that room exists for honest disagreement. I would think that the publisher of such "original" classics as Missile Defense and Jawbreaker would be well aware that one man's "rip-off" is another man's "free enterprise", and not be so quick to judge. Your refusal to advertise in publications which carry the ads of "known pirates or copyright violators" smacks of the same kind of crude coercion as Atari's move to deter the marketing of what they consider to be "copycat" software by use of the threat of expensive litigation as a club. If indeed these outfits are "known pirates and copyright violators", then sue them and put them out of business, instead of asking us to "rid our newsletter" of their ads. Even Atari does not stoop to this kind of pettiness, although they do not regard you any more highly than do you the publishers of Locksmith.

We are deeply concerned with the problem of software piracy, and the harm it does both to software publishers and consumers. It is clear, however, that the solution does not lie in treating all customers as potential thieves, nor in penalizing the honest consumer with inflated prices to make up for piracy losses. Such practices create resentment among consumers, who look at the software publisher who indulges in them as an adversary from whom they may steal without remorse. The more productive course for the software vendor to take is to offer what the pirate cannot: fair and honest prices, follow-up service, thorough documentation, and a reasonable update policy for software improvements. By gaining the respect of the software-buying public, such a vendor does more to curtail piracy than he could by using any copy-protection scheme. There is no known device which can stand up for very long against someone with a little technical skill, and the will to defeat it.

We hope that this clarifies MACE's position, and that you will



again consider our newsletter as a possible vehicle both for advertisement, and for expression of your opinions on issues which affect the relationship between software vendors and their customers. Given the strong mutuality of our interests, it makes much more sense to resolve any differences we may have through dialogue than through boycotts.

I would also like to take this opportunity to let you know of two areas of your operation in which I believe there is room for improvement. The first is your insistence on spelling the word "escalating" as "esculating" in your ad copy. The second is the magazine which you send to registered owners of your Atari software. I for one do not believe that you are demonstrating any real commitment to serving the needs of Atari owners by regularly sending them a couple hundred pages of advertising for the Apple.

Very truly yours,

Sheldon Leemon,  
Secretary

---

## MACE Proprietary Programs

MACE proprietary programs are authorized editions of commercial software made available by special arrangement with the authors. They are available to MACE members only at general membership meetings for nominal media and documentation fees.

MACE SMART TERMINAL Version 2.1 ... by Tom Giese  
Simply the best smart terminal program for the Atari.  
Version 2.1 features include:

- Smart Upload/Download
- Redefinable Translation Tables
- All 6502 Assembly Code
- Multiple Baud Rate Support  
(including Bell 212A)

"More Bells and Whistles Than A Mississippi River Boat!!"

Ian Butterworth  
Noted Datacom Heavy

All program parameter defaults can be reset by the user. Why insult your SmartModem by using a dumb terminal program? Do ANY modem you use with your Atari and 850 interface a favor... throw out your Telelink I and start using the MACE Smart Terminal. Only \$6.00 (Owners of the Version 1 release may obtain Version 2.1 for only \$1.00 by returning both their original program diskette/cassette and documentation to the club disk librarian.)

INSTEDIT ... by Sheldon Leemon  
The Award winning program described by Atari as simply

"the best character set editor we have seen". Order it through APX if you wish for \$18.00, or purchase this authorized version as a MACE member for only \$6.00. Voluminous and concise documentation.

"If I ever write a program you can bet I'll use INSTEDIT to civilize my character set."

Arlan Levitan

## DECEMBER MEETING MINUTES

Sheldon Leemon, Secretary

In the spirit of the upcoming holiday season, this meeting previewed some of the software presents we may soon be getting (even if we have to buy them for ourselves). For the adventurer, ALI-BABA offers multiple player madness, with everyone participating at once. The new high-res WIZARD AND THE PRINCESS augments the usual text-only format with some very nice pictures. For the pool lovers among you, POOL 1.5 lets you replace the stick with the paddle for some felt-less fun. This is an extremely well done simulation, with nice play options.

For the business portion of our meeting, we discussed the possible committees that could be formed to allow for a little closer interaction within the group. Next meeting we will be voting to adopt the more popular selections.

Our main speaker was Chuck Miller of K-BYTE, a local software developer. He gave us a mercifully abbreviated view of the trials and tribulations faced by those who try to bring a software idea to market. He also gave us a preview of his first offering, K-RAZY SHOOT-OUT, a fast-action arcade game that will soon be available in an 8K ROM pack (much to the delight of those 400 owners). Its very professional, and extremely addictive. Hopefully, a full review will appear in these pages.

The Executive Board would like to extend holiday greetings to all members, and to remind you that a MACE membership makes the perfect gift for you or someone else you like. Unlike Santa, MACE keeps bringing you presents all year. We've already brought you the MACE library, featuring commercial software developed by MACE members and offered at a fraction of their store price. Our monthly meetings feature fascinating speakers to delight the expert and novice alike. We've offered you BASIC programming classes. Not to mention the inimitable MACE newsletter. How could you face '82 without MACE? I mean, you might as well own an Apple. Renew now.

WE'RE  
CHANGING THE  
WAY  
DETROIT THINKS  
ABOUT  
COMPUTER  
RETAILING

**BINARY**  
**548-0533**

- \* COMPETITIVE PRICES
- \* FULL SERVICE
- \* KNOWLEDGEABLE STAFF
- \* PRODUCT ASSORTMENT
- \* PROFESSIONAL ASSISTANCE
- \* TRAINING FACILITY
- \* CONTINUING SUPPORT

## Let's Talk Business on the ATARI

Introducing...  
A complete  
business system  
designed for the



✓ Standardizes all of your invoices, statements, purchase orders, credit memos and more.

✓ Detailed account aging reports that show 30-60-90 and over 90-day transactions.

✓ Checks issued automatically with detailed check register.

✓ Accounts Payables package makes it easy to keep track of your obligations.

✓ Bargraphs compare (in full color) income vs. expenses.



Integral Data Systems

**PAPER TIGER**



C. ITOH

**EPSON**

**OKIDATA**

**SOFTWARE**

**DISKETTES**

•Ribbons •Print Wheels •Data Binders  
•Diskette Storage Items •Thermal And  
Printer Paper •Labels •Anti-Static Mats

### COMING SOON!!!

- \*Hi-Res Wizard & the Princess
- \*Adult Adventure Games
- \*Christmas Music
- \*BYTE \$299.00 132 col. printer

### ATTENTION SOFTWARE AUTHORS!!!

Don't Hide Your Work.  
With our help you can  
release a marketable  
product and make some  
money. DO IT NOW.

V I U O  
S S D  
I A  
T Y

NEW HOURS: MON-FRI 9:30-6:00  
SATURDAY TILL 5:00

**BINARY**

COMPUTER  
SYSTEMS

3237 WOODWARD AT 12 MILE  
BERKLEY 548-0533



**THE GAMES WE PLAY****Software Review: K-RAZY SHOOT-OUT**

With the introduction of K-razy Shoot-Out, a new firm called K-BYTE has made an impressive entry into the Atari computer software market. Not only have they produced a very professional animated arcade game, but they have also become the first company outside of Atari to offer software of this type for the 400/800 in ROM-cartridge format. Despite their relatively high price (K-RAZY SHOOT-OUT retails for \$49.95), ROM-based games are very appealing to the consumer market, as they are reliable, durable, and easy to use, even for the minimum-configuration system owner who has relatively little computer experience. As Atari has been actively cultivating consumers, and has been using such games to draw into the market people who have previously been somewhat intimidated by computers, the appearance of second-source ROM packs is a significant event.

In the game of K-RAZY SHOOT-OUT, your on-screen alter ego is a Space Commander whose mission is to rid Alien Control Sectors of the deadly Droids. In each of the maze-like Sectors, the Commander must destroy all Droids before the moving time bar disappears. If he is successful, he progresses to a new and even more hostile Sector. Bonus points are awarded if the job is finished before the time bar changes colors.

Droids can be eliminated by use of a laser weapon, or by maneuvering the mindless drones into colliding with the charged barrier walls, or each other. But collision with the barriers or Droids is also fatal to the Commander, and after the first level, the Droids return laser fire. A Commander who falls in battle is replaced by one of two backups. After the last one is gone, the game is over. The game will also end if the Commander has failed to leave the sector before the time bar disappears. In either case, the player's performance is evaluated according to a complex formula based on the number of Droids destroyed, the Sector level reached, and the time required, and a rank is awarded.

It is evident that the producers of this game have spent a lot of time working on the play values. While the first level can be mastered after a little practice, each new level introduces a quantum leap in difficulty, as the Droids get smarter, faster, and more numerous. The player's skills will gradually improve, but there is little danger of the game becoming so easy that he will lose interest. Part of the reason for this is the random generation of Sector configurations, and random points of entry for replacement Droids (there are only three on screen at any one time). Because time is weighted heavily in the rankings, and can add bonus points to your score, even when you have mastered the basic skills required for reasonable proficiency, there is always room for improvement in speed of play. Taken as a whole, these elements give the game an extremely addictive quality, which is, after all, what arcade games are all about. I think that it is fair to say that even



after the initial excitement wears off, this is a game that you would want to go back to in the future.

Another area in which much care has been taken is human engineering. All play functions are performed with the joystick, so there is no need to use the keyboard. Because of the pattern of action, the player must keep moving the stick in different directions, which helps to prevent his hand from turning into a useless claw after and extended period of play. The figure of the Commander changes color to indicate when there are only six more Droids left in a round. There is a recap at the end of each round, which gives the player a chance to relax and regroup. Also, the game can be restarted at any point by hitting the Start key. This saves a lot of frustration when you are trapped in an impossible Sector.

This is not to say that there isn't room for improvement. The graphics and sound effects are good, but not spectacular. Although the animation of the Commander and Droids is well done, the Sectors themselves are very plain. The ranking system shows no mercy. Even if you struggle along to an upper level, you may find that your rank has dropped back to Goon Class 5 (the lowest level) because you took too much time, or got too few points along the way. This continuing escalation of difficulty may be discouraging to those not hooked on arcade games. An option which would allow you to skip the first couple of sectors would be a terrific addition. As things stand, you've got to play through all of the lower levels in order to get to the one which is currently driving you K-razy. Multiple players would also be a very nice option to have. Unfortunately, there is only so much ROM available in the cartridge.

Despite its limitations, this program serves as a very nice introduction for K-BYTE. Atari computer owners may be extremely encouraged by the direction that ROM-based software for their machine appears to be taking.

K-razy Shoot-Out is available nationally through local dealers, or it can be ordered directly from K-BYTE, 1705 Austin, Troy, Michigan 48099.

---

#### EDUCATIONAL SOFTWARE NOTES from Craig Chamberlain

*Several good articles of interest to the educational software designer have appeared recently. I would like to thank Judy Braun for bringing to my attention two articles in the November 1981 issue of TEACHING Exceptional Children. The first is titled Computers Are for Kids: Designing Software Programs to Avoid Problems of Learning. The author discusses problems that a student might encounter while using a computer program, such as too much typing being required and difficulty in reading complicated instructions. Methods of promoting motivation are also given. The other article is Microcomputers: Powerful Learning Tools with Proper Programming. This one is jam packed with suggestions for evaluating software. Main subject areas covered include Learner/Teacher Needs, Instructional Integrity, and Technical Adequacy and Utility. The material contained in these two articles is excellent. Knowing what a teacher is looking for when examining software will greatly improve software quality.*

continued

Don't laugh when you hear this next one. In the December issue of BYTE there is an article that talks about What Makes Computer Games Fun? Although the topic is computer games, I think that the information will apply just as well to educational programs by explaining how to keep children interested. In other words, the first two articles tell what the teacher is looking for in a program. This one tells what the child is looking for. Game features of the most significance were that the game had a goal, that the computer kept score, that there were audio effects, and that there was a random factor. The most popular kind of game was the graphic game, rather than math or word games (what did you expect?)

In fact, all three of these articles should be read by anyone who plans on writing any kind of program. Can I give a better plug than that?

Finally, ATARI PILOT has been released and at first glance it appears to be very good. The name of the language is an acronym for Programmed Inquiry for Learning Or Teaching. The purpose of the language is to be easy to learn so that young children can quickly start writing their own programs, and teachers can write additional programs for the students. The main strength of PILOT lies in its method for accepting a user's answer to a question and the way that answer is compared with expected answers to find a match. On the other hand, PILOT is rather limited in some respects and was definitely not designed for persons wanting to do complex technical programming such as can be done in BASIC.

The ATARI implementation of PILOT has some nice features. These features include full use of standard ATARI editing keys, automatic line numbering, line renumbering, string variables, and turtle graphics. Turtle graphics refers to a graphing procedure in which new points are plotted relative to old points, instead of using X and Y co-ordinates of the Cartesian system. So, you can tell the computer to TURN 90 which means that the new drawing direction will be ninety degrees clockwise from the old direction, and then say DRAW 10 which draws a line ten units long in the new direction. ATARI PILOT supports both the turtle and Cartesian systems.

The documentation is superior. Two manuals are included, one is the PILOT PRIMER, the other is the Reference Guide. The primer assumes that the reader doesn't know what a computer is, and from there in ten chapters does a real nice job of educating the reader in the use of the language. Each chapter presents new concepts with examples, gives a summary, and then finishes with a little quiz. Important notes are color coded. The Reference Guide, affectionately called Student Pilot, is full of cute illustrations and examples so young people will be able to use the guide with ease. For example, the SAVE operation is pictured by showing program characters walking from the computer to a program recorder. A beginning pilot, his pet turtle, and an ATARI computer unmistakably named Fred accompany the reader through the whole guide.

Both manuals are in color on slick enameled paper and constitute a total of three hundred pages of information. The package also includes a pocket reference guide, left slot cartridge, and two demonstration cassettes which have their own users guide. It should be mentioned that ATARI PILOT can also use the disk drive and DOS. All of this comes in a nice binder and sells for \$89.95. It's not for everybody but will be fantastic for educators. This release by ATARI will certainly interest a lot of people.

*Craig Chamberlain*

12/23/81

## BAKER STREET BYTES

By Richard Gizynski

Last month, we discussed the way that the DIMensioned space for strings was placed at the end of memory and the two lists that are kept for the dimensioned spaces. List one tells how much space is dimensioned for each string. List two tells how much space that has been dimensioned is actually being used by an 'active' string.

Those of you that experimented with several strings, probably found out that the strings are kept in the same order that they are DIMed, with no gaps in between.

There are a couple of ways to 'open up' a DIMed string. The first and most common is to simply write:

```
STRING$="OPEN UP"
```

This opens up the string to the number of characters that were used. The second way of opening a string is similar. You simply:

```
STRING$(last DIMed number ,  
last DIMed number)=" "
```

This will open up the full string to be used. A third way of opening up a string is to make the string equal to a string that has already been opened and is as large as you need or larger:

```
STRING$=STRING2$
```

The forth way is to look up the string table address by PEEKing into 130(low) and 131(hi)\*256 and finding the address of the table that holds the string names and then modifying that table.

Now, on to useful things. First, if your are going to use the string for holding text information, its handy to have a clean string. Short strings are cleaned by putting a lot of spaces between an open and closed set of quotes. That way, when you sent the string to the screen or printer, the part of the string that you didn't use will appear blank on the screen. Larger strings can be 'cleaned':

```
10 REM TELEPHONE NUMBERS
20 REM BY RICHARD GIZYNSKI
30 REM
40 PRINT "}"
50 ? "ONE MOMENT PLEASE"
60 SIZE=INT((INT(FRE(0)+0.7)/38)*38)
70 DIM CLEAN$(38),MAINSTRING$(SIZE),NAME$(38)
75 DIM NUMBER$(38),LINE$(38),WORK$(7)
80 REM STRING CLEANING
85 REM LINE 90 FOURTY SPACES
90 CLEAN$=""
100 NAME$=CLEAN$
110 NUMBER$=CLEAN$
120 LINE$=CLEAN$
130 FOR I=0 TO SIZE/38-1
140 MAINSTRING$(I*38+1,I*38+38)=CLEAN$
150 NEXT I
160 ENTRY=0
170 GOTO 970
```

This not only cleans the strings but opens them up.

First, lets create a way to input some information that we can work with:

```
180 REM INPUTING DATA FROM KEYS
190 PRINT "}"
200 ? "TO GOTO MENU JUST HIT RETURN WITHOUT A NAME"
210 NAME$=CLEAN$:NUMBER$=CLEAN$:LINE$=CLEAN$
220 ? "INPUT NAME AND PRESS RETURN":INPUT NAME$
230 IF NAME$="" THEN GOTO 970
240 ? "INPUT TELEPHONE NO. AND PRESS RETURN":INPUT NUMBER$
```

For those of you who would like variable names to be kept temporarily, the cleaned strings will probably do the job. By taking a long string and concatenating it, you can have a form of string array. That is:

```
300 REM CONCATENATING A STRING
310 LINE$=WORK$
320 LINE$(5)=NAME$
330 LINE$(39-LEN(NUMBER$))=NUMBER$
340 MAINSTRING$(ENTRY*38+1)=LINE$
350 IF ENTRY<INT((SIZE/38)-1) THEN GOTO 180
360 ? "MY MEMORY IS FULL"
370 FOR I=1 TO 200:NEXT I:GOTO 970
```

will allow you to keep the names of many items in the above DIMed string MAINSTRING\$. From this point on, all you need to keep track of is the number that you associate the name with.

Another handy thing that can be done with strings is printing out large numbers with commas, numbers with two decimal numbers or numbers with special purposes such as a telephone number. Remember when setting up a number in a string that Atari will not read a number beyond a comma or a character, but it will read a number beyon. J a period. Atari treats periods and decimal points the same.

In the case of some numbers, we would want Atari to align the number on the right hand side:

```
250 REM MAKING NUMBERS FLUSH RIGHT
260 ENTRY=ENTRY+1
270 WORK$=STR$(ENTRY)
280 IF ENTRY<100 THEN WORK$=" ":WORK$(2)=STR$(ENTRY)
290 IF ENTRY<10 THEN WORK$=" ":WORK$(3)=STR$(ENTRY)
```

Printing out the string to the screen or printer becomes very easy with this type of string. Note the semicolon (on line 400) at the end of the MAINSTRING\$ paren. It keeps the 38 character segments printing on adjacent lines on the screen. Without it there would be a blank line between the printed lines. Since all printers are at least 40 character wide, don't use the semicolon at the end of line 460.

*continued*

```

380 REM PRINTING ON THE SCREEN
390 FOR I=1 TO ENTRY
400 ? MAINSTRING$(I*38+1,I*38+38);
410 NEXT I
420 ? "PRESS RETURN FOR MENU"
430 INPUT WORK$:GOTO 970
440 REM PRINTING ON THE PRINTER
450 FOR I=1 TO ENTRY
460 LPRINT MAINSTRING$(I*38+1,I*38+38)
470 NEXT I
480 GOTO 970

```

Now that we have a useful string started, we need to be able to save it. One way of doing this is to PUT the string on tape or disc. This has to be done character by character, after converting the string to its numeric value. But, in order to be able to retrieve it intact later on, we will make the first portion of the string hold the information of how long the string is. That way, when we start to read the string back, we'll have a way of telling how long it is and how much to read back.

```

490 REM SAVING TO TAPE/DISK
500 PRINT "":POKE 84,6:CLOSE #2
510 ? "WOULD YOU LIKE TO SAVE TO":? "TAPE OR DISC +T/D/MENU+"
520 INPUT WORK$
530 IF WORK$(1,1)="T" THEN OPEN #2,8,0,"C":GOTO 610
540 IF WORK$(1,1)="D" THEN GOTO 570
550 IF WORK$(1,1)="M" THEN 970
560 GOTO 500
570 ? :? "ENTER 1-8 CHARACTER NAME ":INPUT NAME$
580 IF LEN(NAME$) (1 OR LEN(NAME$))8 THEN 570
590 LINE$="D1:":LINE$(4)=NAME$:OPEN #2,8,0,LINE$
600 GOTO 620
610 ? "POSITION TAPE AND PRESS RETURN"
620 LENGTH=LEN(MAINSTRING$)
630 WORK$=STR$(LENGTH)
640 MAINSTRING$(1,5)=WORK$
650 WORK$=STR$(ENTRY)
660 MAINSTRING$(6,8)=WORK$
670 FOR I=1 TO LEN(MAINSTRING$)
680 N=ASC(MAINSTRING$(I))
690 PUT #2,N
700 NEXT I
710 CLOSE #2
720 GOTO 970
730 REM INPUT FROM TAPE/DISK
740 PRINT "":POKE 84,6:CLOSE #2
750 ? "DO YOU WANT INPUT FROM"
755 ? "TAPE/DISC OR RETURN TO MENU (T/D/MENU)"
760 INPUT NAME$
770 IF NAME$(1,1)="M" THEN 970
780 IF NAME$(1,1)="T" THEN DEVICE=1:GOTO 840
790 IF NAME$(1,1) (<) "D" THEN 740
800 ? :? "ENTER 1-8 CHARACTER NAME ":INPUT NAME$
810 IF LEN(NAME$) (1 OR LEN(NAME$))8 THEN 800
820 LINE$="D1:":LINE$(4)=NAME$:OPEN #2,4,0,LINE$
830 GOTO 860
840 ? "POSITION TAPE AND PRESS RETURN"

```

```

850 OPEN #2,4,0,"C:"
860 FOR I=1 TO 5
870 GET #2,N
880 WORK$(I)=CHR$(N)
890 NEXT I
900 FOR I=6 TO VAL(WORK$)
910 GET #2,N
920 MAINSTRING$(I)=CHR$(N)
930 NEXT I
940 CLOSE #2
950 ENTRY=INT(LEN(MAINSTRING$)/38)-1
960 GOTO 970

```

When writing a complex program,  
it is often handy to drive it from a menu:

```

970 REM MENU
980 REM
990 TRAP 970
1000 ? "":POKE 84,4:POKE 82,10:
1010 POKE 85,15: ? " MENU ":
1020 ? "1. INPUT FROM KEYBOARD"
1030 ? "2. INPUT FROM TAPE/DISC"
1040 ? "3. SAVE DATA"
1050 ? "4. ? DATA ON PRINTER"
1060 PRINT "5. ? DATA ON SCREEN"
1070 ? "6. INSTRUCTIONS"
1080 ? "7. EDIT DATA"
1090 ? "8. ALPHABETIZING"
1100 ? :PRINT "CHOOSE ONE OF THE ABOVE"
1110 POKE 82,2:
1120 POKE 85,10:INPUT A
1130 ON A GOTO 180,730,490,440,380,1620,1140,1340

```

There is always of problem of errors in INPUTing data. With a fixed length string, corrections are not difficult. We simply pull out the part of the string that we want to fix, make a correct copy of it, check the copy, and put it back into the MAINSTRING\$:

```

1140 REM EDIT DATA
1150 PRINT "":TRAP 1140
1160 ? "ENTER THE NUMBER OF THE":? "ITEM YOU WISH TO CHANGE"
1170 ? "IF YOU WISH TO GO TO MENU ENTER 0"
1180 INPUT ITEM:IF ITEM=0 THEN 970
1190 LINE$=MAINSTRING$(ITEM*38+1)
1200 ? LINE$
1210 ? :? "NOW INPUT THE CHANGED NAME":INPUT NAME$
1220 ? "NOW INPUT THE CHANGED NUMBER":INPUT NUMBER$
1230 LINE$=CLEAN$
1240 IF ITEM(10 THEN WORK$=" ":WORK$(3)=STR$(ITEM)
    IF ITEM(100 AND ITEM)9 THEN WORK$=" "
    :WORK$(2)=STR$(ITEM)
1250 IF ITEM)99 THEN WORK$=STR$(ITEM)
1270 LINE$=WORK$:LINE$(5)=NAME$:LINE$(39-LEN(NUMBER$))=NUMBER$
1280 ? :? "IS THIS CORRECT?"
1290 ? :? LINE$
1300 INPUT WORK$:IF WORK$(1,1)="N" THEN GOTO 1140

```

*continued*



```
1310 IF WORK$(1,1)("&Y" THEN GOTO 1280
1320 MAINSTRING$(ITEM*38+1,ITEM*38+38)=LINE$
1330 GOTO 970
```

Most good list should be alphabetized. This requires an ability to sort through the MAINSTRING\$ and move the segments to there proper order. This is done by comparing the first segment of MAINSTRING\$ to the second segment. If the first segment is larger, that is has a character that appears later in the alphabet, then it is pulled out and the second segment replaces the first. Atari compares letters as

```
1340 REM ALPHABETIZING
1350 GRAPHICS 18:SETCOLOR 4,4,1:SETCOLOR 0,8,7
1360 POSITION 6,5: ? #6;"WORKING"
```

though they were digits in a number. That is, the first character of a string is the "most significant" character. Then Atari considers the second character. In the example below, I bypassed the part of the string containing the entry number. Entry numbers are used to help locate the substring for editing and manipulating.

```
1370 WORK$="WORKING"
1380 IF WORK$="DONE" THEN GOTO 1530
1390 WORK$="DONE"
1400 FOR I=1 TO ENTRY-1
1410 CYCLE=CYCLE+1
1420 FS=I*38+1:SS=(I+1)*38+1
```

```
1430 FSL=FS+4:SSL=SS+4
1440 IF MAINSTRING$(FSL,FSL+10)MAINSTRING$(SSL,SSL+10)
    THEN 1460
1450 GOTO 1510
1460 LINE$=CLEAN$
1470 LINE$=MAINSTRING$(FS)
1480 MAINSTRING$(FS,FS+37)=MAINSTRING$(SS,SS+37)
1490 MAINSTRING$(SS,SS+37)=LINE$
1500 WORK$="WORKING"
1510 NEXT I
1520 GOTO 1380
```

By changing the +10 in line 1440 to another number you can sort out by more or less characters. Or by adding a comma and a second parameter to the first MAINSTRING\$ parameter, you can have Atari sort out and list by telephone numbers or any other feature.

Next, we want to make the entry numbers agree with where the entries are:

```
1530 REM NEW LINE NUMBERS
1540 FOR I=1 TO ENTRY
1550 IF I<10 THEN WORK$=" ":WORK$(3)=STR$(I)
1560 IF I>9 AND I<100 THEN WORK$=" ":WORK$(2)=STR$(I)
1570 IF I>99 THEN WORK$=STR$(I)
1580 MAINSTRING$(I*38+1,I*38+3)=WORK$
1590 NEXT I
1600 GRAPHICS 0
1610 GOTO 970
```

*continued*

## COMPUTER CENTER

The Computer Store for EveryOne

**West Bloofmield**  
4381 Orchard Lake Road

**855-4220**



**Garden City**  
28251 Ford Road

**425-2470**  
or **422-2570**

# ATARI™

Last, but not least, instructions are helpful. This section can easily be omitted or condensed to save part of the 2.2K that it takes up. I rushed through this part a little. If you take out this section, don't forget to remove it from the menu and the 1620 figure from line 1130. Otherwise you would have an error. If you forget to remove it, I've TRAPed the program at line 990 to return you to the MENU anyway. But it is frustrating to GOTO a section that isn't there.

This program is only a starting point. You can add a lot of features to it, compact it, improve its running speed etc. If you have any good additions to it please let us know. You might also like to use this string handler for lots of other things.

```

1620 REM PRINT INSTRUCTIONS
1630 REM OPTIONAL SECTION. IF YOU LEAVE
1635 REM IT OUT, REMEMBER TO CHANGE MENU
1640 PRINT "}"
1650 ? "THIS IS AN EASY TO USE TELEPHONE"
1660 ? "DIRECTORY. YOU START BY GOINT TO MENU"
1670 ? "IF YOU HAVE ALREADY STORED DATA ON"
1680 ? "TAPE OR DISC, BRING THIS INTO THE"
1690 ? "PROGRAM FIRST. THEN YOU CAN ADD TO IT"
1700 ? "BY USING THE MENU OPTION 'KEYBOARD'"
1710 ? :? "'KEYBOARD' INPUT IS FOR STARTING OR"
1720 ? "ADDING DATA TO THE LIST. IT IS MOSTLY"
1730 ? "SELF PROMPTING. NAMES SHOULD BE ADDED"
1740 ? "LAST NAME FIRST, THEN A COMMA IF YOU"
1750 ? "WISH OR A SPACE, THEN PRESS THE RETURN"
1760 ? "IF YOU MAKE A MISTAKE, YOU CAN CORRECT"
1770 ? "IT BEFORE YOU PRESS THE RETURN OR IN"
1780 ? "THE EDIT PORTION OF THE PROGRAM. THEN,;"
1790 ? "ENTER THE PHONE NUMBER AND PRESS"
1800 ? "RETURN. THE ONLY LIMIT IS YOU MAY ONLY;"
1810 ? "USE 33 CHARACTERS FOR THE NAME AND NO.;"
1820 ? "YOU MAY USE LESS. SPACES ARE ADDED"

```

```

1830 ? "AUTOMATICALLY":? "PRESS RETURN TO CONTINUE"
1835 INPUT WORK$
1840 PRINT "}"
1850 ? "THE EDIT FEATURE IS PROVIDED SO THAT"
1860 ? "ANY MISTAKES THAT YOU NOTICE. "?:
1870 ? "TO EDIT, YOU MUST KNOW THE ITEM NUMBER";
1880 ? "YOU ENTER THE ITEM NUMBER AND PRESS"
1890 ? "RETURN. THIS WILL BRING UP THE LINE"
1900 ? "TO BE EDITED. YOU ENTER THE CORRECT"
1910 ? "NAME PRESS RETURN. THEN THE CORRECT"
1920 ? "NUMBER AND PRESS RETURN. REMEMBER --"
1930 ? "DO NOT ADD THE ITEM NUMBER. IT WILL"
1940 ? "BE ADDED AUTOMATICALLY. THEN THE PRO-"
1950 ? "GRAM WILL ASK IF THE LINE IS CORRECT"
1960 ? "AND SHOWS YOU THE LINE. IF YOU SAY YES";
1970 ? "THE LINE IS ENTERED IN THE STRING AND"
1980 ? "YOU ARE RETURNED TO MENU. IF YOU SAY"
1990 ? "NO YOU ARE RETURNED TO THE SAME "
2000 ? "SECTION. BY ENTERING 0 WHEN ASKED FOR"
2010 ? "AN ITEM NUMBER YOU'LL GO BACK TO MENU"
2020 ?
2030 ? "PRESS RETURN TO CONTINUE":INPUT WORK$
2040 ? "}"?: "AFTER ENTERING FROM THE KEYBOARD, YOU"
2050 ? "YOU WILL WANT TO ALPHABETIZE THE LIST"
2060 ? "SIMPLY ENTER 6 AND THIS WILL BE DONE"
2070 ? "AUTOMATICALLY. YOU WILL SEE A GRAPHICS";
2080 ? "CHANGE. THIS IS TO ALLOW PROCESSING"
2090 ? "TO TAKE PLACE A LITTLE FASTER":?
2100 ? "THE PRINTER SELECTION PRINTS THE"
2110 ? "STRING IN 38 CHARACTER FLUSH LEFT AND"
2120 ? "RIGHT FORMAT. SIMPLY ENTER THE MENU"
2130 ? "SELECTION."
2140 ? "PRESS RETURN FOR MENU":INPUT WORK$
2150 GOTO 970

```

## UNCLASSIFIED ADS

16K RAM... \$65.00  
Pat McCabe, 791-0946

TI-743  
SILENT 700 PRINTER TERMINAL  
SID CHODUN, 559-1639

## M.A.C.E Newsletter Advertising Policy

Short, 1 line ads will be printed for members at no charge.

Commercial ads will be printed for the following rates:

¼ Page — \$15      ½ Page — \$25      Full Page — \$50

All ads MUST be submitted by the first of the month, be photocopy ready, and be accompanied by a check for the proper amount.

## QUESTIONS AND ANSWERS

THE QUESTION AND ANSWER BOX by Craig Chamberlain

1. How do you make a cassette BASIC program that automatically runs after you load it?
2. How do you make a BASIC program automatically restart after the SYSTEM RESET key is pressed?

The answer to these two very good questions lies in modifying the CIO E: (editor) device, a technique which will be explained in a future article in the newsletter.

3. If you modify the display list to have multiple screens, how do you tell CIO which one to use?

Whenever CIO calls the screen handler to open a new graphics mode, the fifth and sixth bytes of the new display list, the LMS (load memory scan) bytes are saved in SAVMSC (\$0058,2 or decimal 88 and 89). Location SAVMSC is always referenced whenever data is written to or read from the screen. Whereas the display list tells ANTIC where display memory starts, SAVMSC tells the computer where it starts. You can manually change either address, so it is possible to display one screen while writing to another. Sheldon Leemon may write an article on this topic for future publication.

4. How can you read data from the cassette?

The normal CIO device calling procedure also applies to cassette, so the OPEN, GET, PUT, CLOSE and STATUS commands should work. More information will be given in a future article about cassette I/O.

5. How do you POKE tab settings?

Since one logical line can contain a maximum of 120 characters (three screen widths of forty characters each), there are 120 possible tab settings. A tab setting can be considered a binary entity (at any given location there either is a tab setting or there isn't) so it could be represented by a binary bit. Dividing 120 locations by 8 bits per byte gives 15 bytes required. These fifteen bytes can be found starting at TABMAP (\$02A3,15 or decimal 675 through 689). Each byte is initialized to a value of one during power up, SYSTEM RESET, and screen OPEN calls. Add multiples of two to the appropriate bytes to insert new tab settings.

6. How do you get ATARI PILOT to use graphics modes other than mode seven, and how do you change the colors?

The demonstration programs show how to do those things. The normal CIO calling procedure is not used and I haven't yet figured out exactly what they are doing. Hopefully we will have more on this at a later time.

7. Give a summary of player missile graphics.

You gotta be kidding! So many articles about this have already been written that it would be best for me to simply tell you where you can find them. The DE RE ATARI manual presented the philosophy behind this graphics system; you can read the excerpt in November 1981 BYTE if you haven't gotten the manual yet. The first issue of the ATARI CONNECTION gave a simple program to get you started. The January 1981 COMPUTE! had an article that went into more detail. The HARDWARE MANUAL gives the function of each bit in the related hardware registers. And, our very own MACE club plans to have a seminar on players and missiles, so be on the watch for further details.

8. You talk so much about CIO. WHAT IS CIO???

Yes, I do talk too much. The main operating system of your ATARI computer, one of the things which makes the ATARI superior to other microcomputers in its price range (most other micros don't even have a true operating system), the key to understanding the system, the thing which makes it all work, is called the Central Input/Output utility. And I suggest you read Bill Wilkinson's series in COMPUTE! so you can learn more about it.

Enough typing - I have homework to do.

*Craig Chamberlain*

12/24/81

```
20 REM * COLORS
30 REM By Judy Braun
40 REM
50 REM THIS IS AN EXAMPLE OF
60 REM GETTING DIFFERENT COLORS
70 REM EASILY IN GRAPHICS 2+16
75 REM
80 GRAPHICS 2+16
90 POSITION 1,2
100 ? #6:"CAPITAL LETTERS"
110 FOR T=1 TO 1000:NEXT T
120 POSITION 1,4
130 ? #6:"lower case letters"
140 FOR T=1 TO 1000:NEXT T
150 POSITION 1,6
152 REM
155 REM Before typing what's inside
156 REM the quotes in line 160, push
157 REM the ATARI key once.
158 REM
160 ? #6:"INVERSE CAPITALS"
170 FOR T=1 TO 1000:NEXT T
180 POSITION 1,8
184 REM
185 REM Before typing what's inside
186 REM the quotes in line 190, push
187 REM the ATARI key once and then
188 REM push the CAPS/LOWR key once.
189 REM
190 ? #6:"inverse lower case"
200 FOR T=1 TO 1000:NEXT T
210 POSITION 1,10
214 REM
215 REM 'Numbers' and 5 and 0 are in
216 REM inverse video
217 REM
220 ? #6:"Try Numbers 1 5 9 0"
230 FOR T=1 TO 1000:NEXT T
235 POSITION 1,1
240 ? #6:"Now let's change"
245 ? #6:" the background...."
250 FOR B=0 TO 100
260 SETCOLOR 4,B,6
270 FOR T=1 TO 500:NEXT T
280 NEXT B
290 GOTO 290
```



## WORD SEARCH PUZZLE

H O H B Q C P L A R E H P I R E P R B M  
 G O O E F J K E Y B O A R D X X U Z E X  
 P Z C L H O W E L C Y C A G E M H F M T  
 M F N O I T A L U M I S B S E Q N M A O  
 H V Y O X D R O T C E S K I V S U J R L  
 S F S L K I L A T C O U Y W U O A M F I  
 K B I A C C K C A S S E T T E Z W B N P  
 K A B E D A S Y N C R O N O U S V J I T  
 C E L I B O M W O N S F V R E M A C A E  
 I O X T M E D O M K W C Q C E O M P M U  
 S F S R S Z S K A T I N G B B T U X E W  
 A M I E T R E L I P M O C E A D N T Z O  
 B R Q F R O V E C T O R F K D M U I O G  
 R A D F I S K C U B A G E M S L Y P W O  
 I D B U N W L O A Z Z I P G O I Q Q T L  
 T N V B G K V D I T G T Y S T E D K K T  
 S I W R C F N Y R A N I B A W K T S S K  
 L W T P U R R E T N I A Q S L K I Y G K  
 C O A H E X A D E C I M A L C B Q L B K  
 K W H U M P U S Q I A K M N V Z X E U D

## WORD LIST

ABSOLUTE  
 ASYNCHRONOUS  
 BASE  
 BASIC  
 BISYNC  
 BINARY  
 BUFFER  
 BYTE  
 CASSETTE  
 COMPILER  
 DISK  
 HEXADECIMAL  
 ICE  
 INTERRUPT  
 KEYBOARD  
 LOGO

MAINFRAME  
 MEGABUCKS  
 MEGACYCLE  
 MODEM  
 OCTAL  
 PILOT  
 PERIPHERAL  
 PIZZA  
 SECTOR  
 SIMULATION  
 SNOWMOBILE  
 SKATING  
 STRING  
 VECTOR  
 WIND  
 WINTER  
 WHUMPUS

## SUPPORT

**The ATARI® Toll-Free Number.** ATARI provides toll-free telephone numbers that connect you with the ATARI Customer Support Department. Customer Support personnel can give you the name of your nearest ATARI dealer or answer technical questions about your ATARI programs or equipment when you call (800) 538-8547 (in California, (800) 672-1430).

Atari is a reg.™ of Atari, Inc.

## UPGRADE YOUR ATARI®

REPORT CARD

STUDENT *Basic A+*

PARENT *Optimized Systems Software*

SUBJECTS

ADVANCED I/O ..... ☐ A+

STRUCTURED PROGRAMMING ..... ☐ A+

PLAYER/MISSILE GRAPHICS ..... ☐ A+

DEPARTMENT

EASE OF USE ..... ☐ A+

FLEXIBILITY ..... ☐ A+

COMPATIBILITY ..... ☐ A+

THE ONLY LOGICAL UPGRADE FOR ATARI BASIC!

GRADUATE TO BASIC A+  
FROM THE AUTHORS OF ATARI BASIC!

## BASIC A+

BASIC A+ will rate an A+ from any Atari user! Upward compatible with Atari Basic, it adds statements and features that enhance the Atari 800's real power, flexibility, and ease of use: Superior I/O features for business and other applications. Additional file manipulation commands. Significant help in program development and debug. Structured programming aids. And MORE! A partial list of the enhancements of BASIC A+ includes:

RPUT/RGET (record I/O) BPUT/BGET (binary I/O) ERASE  
 PRINT USING SET TAB INPUT "..." DIR PROTECT RENAME  
 TRACE WHILE...ENDWHILE IF...ELSE...ENDIF  
 SUPERB PLAYER/MISSILE GRAPHICS

BASIC A+ requires a disk and 32K bytes of RAM. Since no cartridge is used, BASIC A+ will take advantage of all the RAM (48K bytes) in a maximum Atari 800 system (recommended) ..... \$80

## OS/A+

Completely compatible with Atari's DOS (version 2), but with an advanced, command-driven console processor. Simple. Flexible. Powerful. With an easy-to-use BATCH capability. OS/A+ INCLUDES all the following utilities (and more):

**EASMD** (Editor/ASSEMBLER/Debug) is our upgraded all-in-one assembly language development package for the 6502 microprocessor. The editor — with such features as FIND and REPLACE — can even edit BASIC A+ programs. The assembler can include multiple source files in a single assembly.

**DUPDSK and FORMAT** create master or slave disks. Make sector by sector copies of any OS/A+ disk.

**COPY** a simple, single file copy utility.

All of this power is included in our OS/A+ package ..... \$80

For the utmost in capability and flexibility, our combination system — BASIC A+ and OS/A+ — is available ..... \$150

SOFTWARE IN A CLASS BY ITSELF  
AVAILABLE NOW THROUGH LOCAL DEALERS

Optimized Systems Software, Inc.  
10379 Lansdale Ave., Cupertino, CA 95014  
(408) 446-3099



# H C O M P U T E R H O R I Z O N S

JANUARY IS COMMUNICATIONS MONTH



ATARI 830™ Acoustic Modem

REG 564<sup>95</sup>

SALE 450<sup>00</sup>



ATARI 850™ Interface Module



**THE SOURCE**  
AMERICA'S INFORMATION UTILITY



FREE SEMINARS ON THE SOURCE

JANUARY 16, 1981

10:00  
1:00

CALL FOR RESERVATIONS

(313) 464-6502

37099 SIX MILE ROAD • LIVONIA, MICHIGAN 48152

## ANSWERS TO WORD SEARCH PUZZLE

```

- - - - - L A R E H P I R E P - - -
- - - - - K E Y B O A R D - - - E -
- - C - - - E L C Y C A G E M - - M T
- - N O I T A L U M I S - - E - - A O
- - Y - - - R O T C E S - - - S - - R L
- - S - - I L A T C O - - - - A - F I
- - I - C - - C A S S E T T E - - B N P
- - B E - A S Y N C R O N O U S - - I -
C E L I B O M W O N S - - R - - - A -
I - - - M E D O M - - - - E - - M -
S - - R S - S K A T I N G - - T - - E -
A - - E T R E L I P M O C - - - N T - O
B - - F R - V E C T O R - K - - U I - G
- - - F I S K C U B A G E M S L - - W O
- D - U N - - - A Z Z I P - O I - - L
- N - B G - - - - - - S - E D - - -
- I - - - - Y R A N I B - - - T - - -
- W T P U R R E T N I A - - - - Y - -
- - - H E X A D E C I M A L - - - B -
- W H U M P U S - - - - - - - - -
  
```

## DISCOUNT COMPUTERS

ATARI™ #349

400 w/16K #739

800 w/16K #439

810 Disk Drive #129

32K Board

EPSON #379

MX-70 #479

MX-80 #759

MX-100

ATARI SOFTWARE BIG DISCOUNT  
Jaw Breaker #24.95

ATARI 800 48K #869

RITE WAY ENTERPRISE

Rite Way Enterprise  
P. O. Box 605  
Warren, Mich. 48090

751-2454  
(313) 779-0327

CALL OR WRITE FOR FREE CATALOG

## M.A.C.E. TAPE LIBRARY

### LIBRARY VOL. 1 & 2

#### Side A — Vol. 1

Start Title  
000 Hamurabi  
056 Players & Missles  
080 Search  
160 Computer Graphics  
181 The Price Is Right  
228 Screen Dump  
248 Mailing List  
273 Enemy  
301 Civil War  
358 Fur Trader  
406 Maze  
435 Barricade  
461 Modem/Dixk/Cass. Prgm  
487 Fugue

#### Side B — Vol. 2

Start Title  
000 Battle of Numbers  
024 3D Animation  
046 Fugue  
090 Color & Sound Study  
110 Large Print Hints  
122 Dandy Display  
134 States & Capitals  
184 Miniature Golf  
262 Bridge Builder  
299 Graphics & Color Demo#1  
311 Graphics & Color Demo #2  
323 Bouncing Barney  
350 Weasel — Easel  
365 String Create  
380 Number Slide  
428 Function Machine

### LIBRARY VOLUME 4

#### Side A

Start Title  
000 Color Theatre  
040 Spelling Bee Master Program  
079 Spelling Bee Student Program  
105 D & D Assistant  
148 Ten Gallon Hat  
200 Spelling  
252 Memory Diagnostic  
262 Tape Copy  
271 Renumber (Use ENTER "C:"  
to load then type  
GOTO 19000 & RETURN)

#### Start Title

312 Horse Trader  
331 Vocabulary  
376 Atari Dulcimer  
393 Strato blaster  
413 Ricochet  
448 Baseball  
496 Double Cannon  
513 Maze Search  
529 Trench  
567 Master's Golf

### LIBRARY VOLUME 6

#### Side A — Volume 6

Start Title  
000 Index  
058 Math  
111 Home Loan  
140 note In  
161 Note Out  
174 Bubbsort Num  
188 Screen Reader  
201 Numbers  
216 Ellipse  
228 Alcohol  
243 Calendar  
284 Biorythem  
311 Sec Byt  
379 Hangman  
Side B — Volume 6  
010 Day of the Week  
028 Decidump

056 Disassembler  
100 disk tape  
131 Help  
142 Help.wrt  
157 Hexadec  
172 Hot Stuff  
198 Lister  
210 Modem  
233 Super Computer  
259 Pecker  
267 Renumber (see note)  
316 Help  
325 Help.Dat  
336 Civil War  
384 Clewso  
417 Treasure  
449 Dogbite  
488 Electric Co.  
540 Lander Houston  
569 Star Warp

#### SIDE A-VOL. 8

|                |                    |
|----------------|--------------------|
| 0 INDEX        | 400 SPACEDOD       |
| 50 POLYGON     | 421 STATES         |
| 80 FINE SCROLL | 456 GR.10          |
| 100 GREETING   | 463 COPY           |
| 117 OCTADRAW   | 473 ATARI LOGO     |
| 137 BASEBALL   | 486 FONTEDIT       |
| 212 ATARI DEMO | 510 BARRICADE      |
| 272 CHEM DEMO  | 524 SIMON SAYS     |
| 341 INVADERS   | 536 ERROR MESSAGES |
| 363 ZAXWELL2   | 548 SLIDE          |
| 389 TIME       | 590 CIRCLE         |
|                | 600 RENUMBER       |

### LIBRARY VOLUME 3

#### Side A

Start Title  
000 Lunar Lander  
038 Checkers  
071 Brass Pounder  
161 Number Line  
217 Clock  
300 Nerd  
337 Dodge  
365 Flying Saucer  
393 Bee  
414 Multiplication  
465 Awari  
479 Hats  
516 Gomoko  
537 Maze  
551 Beans  
572 Sumer  
632 Search

#### SIDE B

Start Title  
000 Blackjack  
043 Gunner  
076 Flash  
087 Clock  
101 Restore  
152 Cube  
171 Fast Player Missile  
181 Compose  
198 Messiah  
239 Art  
264 Birthday  
281 Blues  
296 Beethoven  
319 Calories  
355 Clock

### LIBRARY VOLUME 5

#### Start Title

#### Side A

000 Index  
050 Star Merchant  
153 Geoquiz Twenty Questions  
with countries  
226 Simon follow light & Sound  
243 Mille Bornes 1000 Milestones  
302 Hat/Ring political Tr. PET  
342 MOO Numbers Game

#### Start Title

358 Treasure follow the map  
383 Hex Puzzle tr. from APPLE  
404 Famous Sayings Hangman  
tr. APPLE  
444 Spiro Graph (Graphics demo)  
455 Groucho Marx (printer picture)  
465 Banner for the MX80 printer  
475 Pretty Print list BASIC program

### M.A.C.E. VOL. 7

#### Side A

Start Title  
000 Index  
060 Backward  
074 Bow  
086 Chopsticks  
105 Clock.DIG  
142 Ellipse  
156 Etch Sketch  
178 Move A Box  
205 Giggie  
216 Jazz  
238 Logo  
255 Marquee  
275 Oxygene  
286 Puff  
311 Shading  
322 Box Demo.  
333 Circles.Art  
340 Keyboard.Mus  
363 Magic  
380 Rocket  
393 Earthquake  
400 Helicopter

#### SIDE B

Start Title  
010 Help  
023 Cards  
071 Horse  
090 Horse.Dat  
106 Horses  
122 Hdem.Chr  
137 Horsev  
153 Botch  
184 Starwars  
216 XMas Tree  
247 Scroll  
258 Puff  
282 Vegas  
312 Star  
321 Nitemare  
341 Dialogue  
363 Booklist.Dat  
371 bokklist

### M.A.C.E. Propertary Discs:

Smart Terminal Package...

Character Set Editor.....

\* Each \$5.00 \*

Library Cassettes..... 3.00

M.A.C.E. Library Discs...\$4.00

Atari Cassette I/O  
Part #1, OPENing remarks  
by Tom Giese

The cassette recorder, like all Atari I/O devices, must be identified to the operating system before it can be used. This is the function of the OPEN command. The OPEN command links the cassette device to an INPUT/OUTPUT CONTROL BLOCK (IOCB). Some I/O commands include an internal OPEN within the command. These commands are: CSAVE, CLOAD, LOAD, SAVE, RUN, ENTER, and LIST. These commands all OPEN iocb #7 when using the cassette. Before you can communicate to the cassette from your BASIC program, it must be "opened". The syntax of the OPEN command is:

```
OPEN #iocb,aux1,aux2,"C:file"
```

Where:

```
iocb - Input/output control
       block number.
aux1 = 4 - Reading from the
       cassette.
       5 - Reading the tape
           no /RETURN/.
       8 - Writing to the
           cassette.
       9 - Writing to the tape
           no /RETURN/.
aux2 = 0 - Long record gap.
       = 128 - Short record gap.
```

C: - refers to the cassette.  
file - is optional, used only  
for documentation.

Opening the cassette for a read operation causes this sequence of events:

1. The speaker beeps once.
2. You must hit the /RETURN/ key unless opened with no /RETURN/ (mode 5).
3. The motor is started.
4. Ten seconds of leader is read.
5. The motor keeps on running.

Opening the cassette for a write operation causes this sequence of events:

1. The speaker beeps twice.
2. You must hit the /RETURN/ key unless opened with no /RETURN/ (mode 9).
3. The motor is started.
4. Twenty seconds of leader is written.
5. The motor keeps on running.

The cassette motor keeps on running because of an OS "glitch". The motor will not stop until the first variable is given for input (when reading), or the first 128 characters are output (when writing). To avoid problems with this, don't OPEN the cassette until you are ready to transfer data.

## HAVE YOU FORGOTTEN SOMETHING?

If you haven't renewed your MACE membership yet, stop POKEing around. PEEK in your wallet or checkbook for \$15.00, and PRINT your name and address on the handy renewal form on the back PAGE. RESTORE our faith in you by mailing your dues in to the P.O. Box or paying at the next general meeting. Please NOTE that if you didn't pay at least \$15 the last time you PUT your money on the TABLE your membership has definitely expired. SAVE time and trouble by renewing now. IF you have allready taken care of this POINT THEN thanks a lot. BYE.



# READY FOR A REAL CHALLENGE?



It's a battle of wits and fast action, this challenging game in ROM form for \*Atari® 400™ or Atari® 800™ Personal Computer Systems. You're trapped in an Alien Fortress. Your mission is to clear your sector of the Alien Droids and progress to the next sector. The first sector seems easy enough... BUT-WATCH-OUT... as you progress from sector to sector, the Droids become more numerous and aggressive. A great challenge and fun too!

K-RAZY SHOOT-OUT is destined to be a favorite of Atari enthusiasts across the country with features like:

- Solid-state ROM pack
- Seven progressive levels of play
- Every game different
- Bonuses and penalties
- Full-color T.V. graphics and space-age sound
- Full-color instruction booklet
- Millions of variations
- A constant challenge

Now at your Atari Dealer!

[WARNING: Playing the game could be HABIT forming]

## Coming soon:


- K-DOS™
- MATCH-EM™
- K-DATE ORGANIZER™


\*Atari is the Registered Trademark of Atari, Inc.

**FOLD & TEAR**

If not available in your area, send a check or money order directly to:  
**K·BYTE • P.O. Box 455 • 1705 Austin Avenue • Troy, Michigan 48063**  
 Toll Free Phone Orders: **1-800-521-0550** • In Michigan: **(313) 524-9575**  
☐ \$49.95, plus \$1.50 shipping and handling, Michigan residents add 4% sales tax.  
☐ \$1.00 for instruction book only.

NAME \_\_\_\_\_  
 ADDRESS \_\_\_\_\_  
 CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

Please allow 3 weeks for delivery.  
☐  ☐ MasterCard  
 CHARGE CARD NO. \_\_\_\_\_  
 EXPIRATION DATE \_\_\_\_\_



## EDITORIAL

Since this is our very first newsletter of the year, I thought I had FINALLY better get around to doing some kind of editorial! Actually, we have had so many excellent contributions lately, that I hated to sacrifice our (limited) space.

Hope you like our new format! We hope that this format will make the publication somewhat harder and easier to read. In the excuse department however, I have tried wherever possible to keep the margins such that it will facilitate punching for storage in a looseleaf binder. The new format may make this a little difficult, so I shall leave it up to YOU, dear members. What are you comfortable with? Since this is YOUR newsletter, I am open to suggestions on format, and would appreciate hearing from you at any time.

Also in the publications department, you may have heard rumors that we are attempting to collate material for a BEST OF MACE book. As it now stands we still need more material (boy do we ever!) so keep it coming folks! Do not be concerned about your own level of expertise. In an organization of our size (growing bigger daily), SOMEONE will benefit from what you have to say! If you are not an expert typist, don't worry. Submit machine readable copy on disk or tape. These will be returned to you as soon as they are dumped to the printer.

And last but not least...

WE HAVE A NEW AND (hopefully) PERMANENT MEETING PLACE!!

That's right folks! No more sitting on the floor or doing sardine simulations. Our beloved Prez has located us a large roomy meeting place with lots of space and extra rooms for the various groups and activities. It's located in Berkley (map and address on front cover) and I hope to see everyone at the next meeting. We will to publish an annual meeting schedule so you can be well informed in advance. Look for it on the back cover of this months issue.

Well that's it except keep those submissions coming. And remember that we are here to mutually enjoy and benefit from our computer investments. I hope we can do more for you than just provide you with a newsletter each month. We have many members representing many different interests. The potential to learn and enjoy from our members is no less than astronomical! Remember that when you paid your fifteen bucks (and if you haven't-please renew now), that we don't just want your money...we want YOU!

## MICHIGAN ATARI COMPUTER ENTHUSIASTS

President: Arlan Levitan  
12709 Borgman  
Huntington Woods, MI 48070

Programs: Gary Luzier  
22530 Gordon  
St. Clair Shores, MI 48081

Vice Pres.: Jerry Aamodt  
4148 Huhn  
Rochester, MI 48063

Librarians: Sam Findley (Disk)  
8171 Denwood, Apt. #1  
Sterling Heights, MI 48077

Treasurer: Judy Braun  
39281 Eliot  
Mt. Clemens, MI 48043

Rodney Graham (Tape)  
12270 Deming  
Sterling Heights, MI 48077

Secretary: Sheldon Leemon  
14400 Elm  
Oak Park, MI 48237

Newsletter: Marshall Dubin  
2639 Hempstead  
Auburn Heights, MI 48057

---

### PROGRAM LIBRARY SUBMISSION FORM

I wish to submit the following program to the M.A.C.E. program library. I warrant that I have good title to this program, and that it does not infringe upon any copyright. Limited rights are transferred herewith for the use of this program within the membership of M.A.C.E. and its associated Atari Computer organizations.

Program Name \_\_\_\_\_ Submission Media: Tape \_\_\_\_\_ Disk \_\_\_\_\_ List \_\_\_\_\_  
Description \_\_\_\_\_  
Category: Business \_\_\_\_\_ Demonstration \_\_\_\_\_ Education \_\_\_\_\_ Game \_\_\_\_\_ Utility \_\_\_\_\_ Other \_\_\_\_\_  
Minimum System: Memory \_\_\_\_\_ K BASIC \_\_\_\_\_ Assembler \_\_\_\_\_ PILOT \_\_\_\_\_ PASCAL \_\_\_\_\_  
Disk \_\_\_\_\_ Tape \_\_\_\_\_ Interface \_\_\_\_\_ Modem \_\_\_\_\_ Printer \_\_\_\_\_ Light Pen \_\_\_\_\_  
Joystick Controllers \_\_\_\_\_ Paddle Controllers \_\_\_\_\_ Keyboard Cont. \_\_\_\_\_  
Name \_\_\_\_\_ Phone \_\_\_\_\_ Date \_\_\_\_\_

---

### M.A.C.E. MEMBERSHIP APPLICATION

\$15.00 fee enclosed

Name \_\_\_\_\_ Phone \_\_\_\_\_  
Street \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
Company (if applicable) \_\_\_\_\_  
System Description \_\_\_\_\_ Disk/Tape \_\_\_\_\_  
Suggestions \_\_\_\_\_  
I can help with . . . \_\_\_\_\_

Send to: M.A.C.E.  
P.O. Box 2785, Southfield, MI 48037

IMPORTANT DATED MATERIAL  
PLEASE DO NOT DELAY

01/83

BULK RATE  
U.S. POSTAGE  
PAID  
PERMIT #431  
SOUTHFIELD, MI

MICHIGAN ATARI COMPUTER ENTHUSIASTS  
P.O. BOX 2785  
SOUTHFIELD, MICHIGAN 48037

# 1982 MEETING SCHEDULE

|          |    |          |    |
|----------|----|----------|----|
| January  | 21 | July     | 15 |
| February | 18 | August   | 19 |
| March    | 18 | Sept     | 16 |
| April    | 15 | October  | 21 |
| May      | 20 | November | 18 |
| June     | 17 | December | 16 |